

Algoritmo de Bellman-Ford

O algoritmo de **Bellman-Ford** serve para encontrar o **menor caminho a partir de um único vértice origem** para todos os outros vértices de um grafo **mesmo quando há arestas com pesos negativos**.

```
BELLMAN-FORD( $G, w, s$ )
1  INITIALIZE-SINGLE-SOURCE( $G, s$ )
2  for  $i = 1$  to  $|G.V| - 1$ 
3      for each edge  $(u, v) \in G.E$ 
4          RELAX( $u, v, w$ )
5  for each edge  $(u, v) \in G.E$ 
6      if  $v.d > u.d + w(u, v)$ 
7          return FALSE
8  return TRUE
```

Fonte: Introduction to Algorithms; Thomas H. Cormen.

Lógica do algoritmo Bellman-Ford:

1. Inicialize todas as distâncias com infinito, exceto a origem (distância 0).
2. Repita **V-1 vezes**, sendo V o número de vértices do grafo:
 - Para cada aresta ($u \rightarrow v$) com peso w , faça o **relaxamento**:
 - Se $\text{dist}[u] + w < \text{dist}[v]$, então atualize:
 $\text{dist}[v] = \text{dist}[u] + w$
3. Passo adicional (opcional): Verificar se existe **ciclo negativo** no grafo.
 - Verifique se após isso ainda é possível fazer relaxamentos.
 - Se sim, há um **ciclo negativo** (sinal de alerta!).

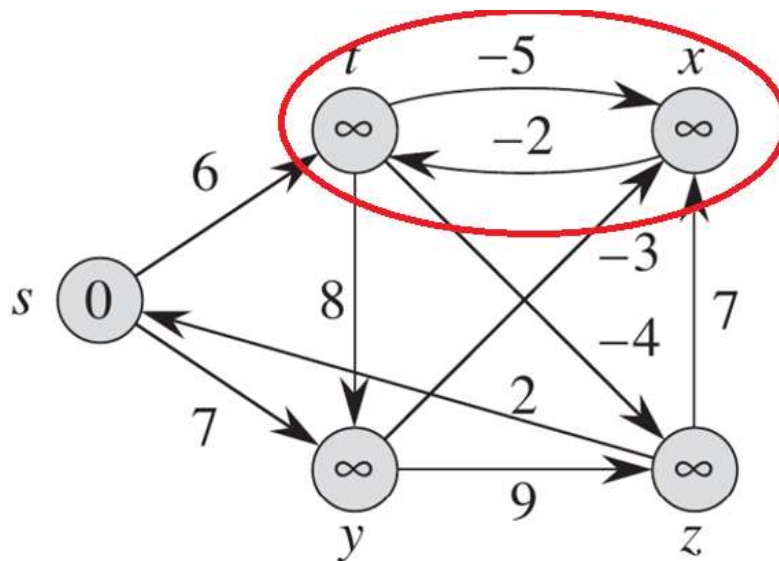
Semelhanças com o Dijkstra:

- Ambos encontram o menor caminho de um vértice origem para todos os outros.
- Ambos utilizam o conceito de relaxamento de arestas.
- Ambos iniciam com uma distância infinita para todos os vértices, exceto a origem.

Diferenças com o Dijkstra:

- Dijkstra não funciona com pesos negativos, enquanto Bellman-Ford funciona.
- Bellman-Ford consegue detectar ciclos negativos; Dijkstra não.

Exemplo de Ciclo Negativo:



Por que ciclos negativos são um problema?

1. Não existe menor caminho bem definido:

- Se há um **ciclo negativo**, é possível continuar passando por ele indefinidamente e reduzir o custo total do caminho a cada volta.
- Em outras palavras: o **menor caminho** até certo vértice pode ser indefinido (e inconsistente) ou tendendo a $-\infty$.

Exemplo: Se um ciclo tem peso total -3, você pode dar 1 volta e reduzir o custo em -3. Se der 100 voltas, reduz -300, o que é irrealista no ponto de vista de uma aplicação prática.

2. Algoritmos como **Dijkstra** e **Bellman-Ford** falham:

- Os algoritmos de Dijkstra e Bellman-Ford assumem que, uma vez que o menor caminho para um vértice foi encontrado, ele não precisa mais ser atualizado.
- Com ciclos negativos, **essa suposição é quebrada**, e o algoritmo pode retornar valores incorretos ou entrar em loop.